

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日            2 0 0 3 年   9 月   9 日  
Date of Application:

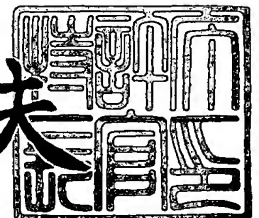
出 願 番 号            特 願 2 0 0 3 - 3 1 7 2 8 2  
Application Number:  
[ST. 10/C]:            [ J P 2 0 0 3 - 3 1 7 2 8 2 ]

出   願   人            株 式 会 社 リ コ ー  
Applicant(s):

2 0 0 3 年   9 月 2 4 日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康 夫



【書類名】 特許願  
【整理番号】 0306271  
【提出日】 平成15年 9月 9日  
【あて先】 特許庁長官 今井 康夫 殿  
【国際特許分類】 G03G 21/00 370  
【発明者】  
    【住所又は居所】 東京都大田区中馬込 1 丁目 3 番 6 号 株式会社リコー内  
    【氏名】 秋吉 邦洋  
【発明者】  
    【住所又は居所】 東京都大田区中馬込 1 丁目 3 番 6 号 株式会社リコー内  
    【氏名】 田中 浩行  
【特許出願人】  
    【識別番号】 000006747  
    【氏名又は名称】 株式会社リコー  
【代理人】  
    【識別番号】 100070150  
    【弁理士】  
    【氏名又は名称】 伊東 忠彦  
【先の出願に基づく優先権主張】  
    【出願番号】 特願2002-307637  
    【出願日】 平成14年 9月14日  
【手数料の表示】  
    【予納台帳番号】 002989  
    【納付金額】 21,000円  
【提出物件の目録】  
    【物件名】 特許請求の範囲 1  
    【物件名】 明細書 1  
    【物件名】 図面 1  
    【物件名】 要約書 1  
    【包括委任状番号】 9911477

**【書類名】 特許請求の範囲****【請求項 1】**

アプリケーションをインストール可能に構成された画像形成装置であって、

アプリケーションを前記画像形成装置にインストールすることが可能か否かを判断するために使用されるアプリケーション情報の取得を行うアプリケーション情報取得手段を備えたことを特徴とする画像形成装置。

**【請求項 2】**

前記アプリケーション情報取得手段は、前記画像形成装置におけるアプリケーションが自ら保有するアプリケーション情報を取得することを特徴とする請求項 1 に記載の画像形成装置。

**【請求項 3】**

前記アプリケーションが自ら保有するアプリケーション情報は、当該アプリケーションの実行ファイル中に含まれる情報であることを特徴とする請求項 2 に記載の画像形成装置。

**【請求項 4】**

前記アプリケーション情報取得手段は、前記画像形成装置におけるアプリケーションとは別のファイルから前記アプリケーション情報を取得することを特徴とする請求項 1 に記載の画像形成装置。

**【請求項 5】**

前記アプリケーション情報取得手段は、前記アプリケーションを仮起動させることにより、前記アプリケーション情報を取得することを特徴とする請求項 3 に記載の画像形成装置。

**【請求項 6】**

前記アプリケーション情報取得手段は、仮起動させたアプリケーションからプロセス間通信により前記アプリケーション情報を取得することを特徴とする請求項 5 に記載の画像形成装置。

**【請求項 7】**

前記アプリケーションは、

前記アプリケーション情報取得手段からの要求に基づき、仮起動するか通常起動するかを判断するステップと、

仮起動の場合に、前記アプリケーション情報取得手段と通信することにより、アプリケーション情報を前記アプリケーション情報取得手段に提供するステップを前記画像形成装置に実行させることを特徴とする請求項 5 に記載の画像形成装置。

**【請求項 8】**

前記アプリケーション情報取得手段が取得したアプリケーション情報を記憶する記憶手段をさらに備えたことを特徴とする請求項 1 に記載の画像形成装置。

**【請求項 9】**

前記記憶手段は、ハードディスク、不揮発性メモリ、およびメモリカードの少なくとも一つであって、

前記アプリケーション情報取得手段は、それらの記憶媒体に記憶された情報を読み出すことによりアプリケーション情報を取得することを特徴とする請求項 8 に記載の画像形成装置。

**【請求項 10】**

前記アプリケーション情報取得手段が取得するアプリケーション情報は、アプリケーションに固有のプロダクト ID を含むことを特徴とする請求項 1 ～ 9 のいずれか一つに記載の画像形成装置。

**【請求項 11】**

前記アプリケーション情報は、前記プロダクト ID に加えて、対応するアプリケーションのベンダー名とアプリケーション名、バージョン情報と使用リソース情報の少なくとも一方を含むことを特徴とする請求項 10 に記載の画像形成装置。

**【請求項 1 2】**

前記記憶手段は、インストールを可能なアプリケーションのプロダクト I D を記憶し、判断対象となるアプリケーションのプロダクト I D がその記憶手段に記憶されているか否かによって、インストール可能なアプリケーションか否かを自動的に判断することを特徴とする請求項 1 0 または 1 1 に記載の画像形成装置。

**【請求項 1 3】**

画像形成処理で使用されるハードウェア資源と、  
ハードウェア資源の制御を行うコントロールサービスと、  
コントロールサービスをサーバとしたクライアントプロセスとして動作し、前記アプリケーションをクライアントとしたサーバプロセスとして動作する仮想アプリケーションサービスを有する請求項 1 ～ 1 2 のいずれか一つに記載の画像形成装置。

**【請求項 1 4】**

前記仮想アプリケーションサービスは前記アプリケーション情報取得手段を含む請求項 1 3 に記載の画像形成装置。

**【請求項 1 5】**

アプリケーションをインストール可能に構成された画像形成装置において使用される方法であって、

アプリケーションを前記画像形成装置にインストールすることが可能か否かを判断するために使用されるアプリケーション情報の取得を行うアプリケーション情報取得ステップを有することを特徴とする方法。

**【請求項 1 6】**

前記アプリケーション情報取得ステップにおいて、前記画像形成装置は、前記画像形成装置におけるアプリケーションが自ら保有するアプリケーション情報を取得することを特徴とする請求項 1 5 に記載の方法。

**【請求項 1 7】**

前記アプリケーションが自ら保有するアプリケーション情報は、当該アプリケーションの実行ファイル中に含まれる情報であることを特徴とする請求項 1 6 に記載の方法。

**【請求項 1 8】**

前記アプリケーション情報取得ステップにおいて、前記画像形成装置は、前記画像形成装置におけるアプリケーションとは別のファイルから前記アプリケーション情報を取得することを特徴とする請求項 1 5 に記載の方法。

**【請求項 1 9】**

前記アプリケーション情報取得ステップにおいて、前記画像形成装置は、前記アプリケーションを仮起動させることにより、前記アプリケーション情報を取得することを特徴とする請求項 1 7 に記載の方法。

**【請求項 2 0】**

前記アプリケーション情報取得ステップにおいて、前記画像形成装置は、仮起動させたアプリケーションからプロセス間通信により前記アプリケーション情報を取得することを特徴とする請求項 1 9 に記載の方法。

**【請求項 2 1】**

前記アプリケーションは、  
要求に基づき、仮起動するか通常起動するかを判断するステップと、  
仮起動の場合に、アプリケーション情報を提供するステップとを前記画像形成装置に実行させることを特徴とする請求項 1 9 に記載の方法。

**【請求項 2 2】**

前記アプリケーション情報を記憶手段に記憶する記憶ステップをさらに有することを特徴とする請求項 1 5 に記載の方法。

**【請求項 2 3】**

前記記憶手段は、ハードディスク、不揮発性メモリ、およびメモリカードの少なくとも一つであって、

前記アプリケーション情報取得ステップにおいて、前記画像形成装置は、それらの記憶媒体に記憶された情報を読み出すことによりアプリケーション情報を取得することを特徴とする請求項 2 2 に記載の方法。

【請求項 2 4】

前記アプリケーション情報は、アプリケーションに固有のプロダクト I D を含むことを特徴とする請求項 1 5 ～ 2 3 のいずれか一つに記載の方法。

【請求項 2 5】

前記アプリケーション情報は、前記プロダクト I D に加えて、対応するアプリケーションのベンダー名とアプリケーション名、バージョン情報と使用リソース情報の少なくとも一方を含むことを特徴とする請求項 2 4 に記載の方法。

【請求項 2 6】

前記記憶手段は、インストールを可能なアプリケーションのプロダクト I D を記憶し、前記画像形成装置は、判断対象となるアプリケーションのプロダクト I D がその記憶手段に記憶されているか否かによって、インストール可能なアプリケーションか否かを自動的に判断することを特徴とする請求項 2 4 または 2 5 に記載の方法。

【請求項 2 7】

アプリケーションをインストール可能に構成された画像形成装置を、  
アプリケーションを前記画像形成装置にインストールすることが可能か否かを判断するために使用されるアプリケーション情報の取得を行うアプリケーション情報取得手段として機能させるプログラム。

【請求項 2 8】

請求項 2 7 に記載のプログラムを記録したコンピュータ読み取り可能な記録媒体。

【書類名】 明細書

【発明の名称】 画像形成装置およびアプリケーション情報取得方法

【技術分野】

【0 0 0 1】

この発明は、コピー、プリンタ、スキャナおよびファクシミリなどの画像形成処理にかかるユーザサービスを提供する画像形成装置で、インストール処理を行うアプリケーションに関するアプリ情報を取得する画像形成装置およびアプリ情報取得方法に関するものである。

【背景技術】

【0 0 0 2】

近年では、プリンタ、コピー、ファクシミリ、スキャナなどの各装置の機能を1つの筐体内に収納した画像形成装置（以下、「複合機」という。）が知られている。この複合機は、1つの筐体内に表示部、印刷部および撮像部などを設けるとともに、プリンタ、コピーおよびファクシミリ装置にそれぞれ対応した3種類のソフトウェアを設け、これらのソフトウェアを切り替えることによって、当該装置をプリンタ、コピー、スキャナまたはファクシミリ装置として動作させるものである。

【0 0 0 3】

このような従来の複合機では、プリンタ、コピー、ファクシミリ、スキャナなどの各機能単位ですべてのアプリケーションプログラムが起動される。これら既存のアプリケーションの場合、各アプリケーションは、起動可能な状態にするためのインストール処理が事前に行われている。すなわち、従来の複合機では、あらかじめ搭載されている既存のアプリケーションの場合、起動可能な状態にするインストール処理を行うにあたって、アプリケーションと複合機とのバージョンの整合性や使用するリソース量などが事前にわかっているため、通常にインストール処理を行って、起動させて実行するだけでよく、インストール前にバージョンやリソース量などをチェックする必要はなかった。

【特許文献1】 特開平11-112701号公報

【発明の開示】

【発明が解決しようとする課題】

【0 0 0 4】

ところで、このような従来の複合機では、プリンタ、コピー、スキャナおよびファクシミリ装置に対応するソフトウェアをそれぞれ別個に設けているため、各ソフトウェアの開発に多大の時間を要する。このため、出願人は、表示部、印刷部および撮像部などの画像形成処理で使用されるハードウェア資源を有し、プリンタ、コピーまたはファクシミリなどの各ユーザサービスにそれぞれ固有の処理を行うアプリケーションを複数搭載し、これらのアプリケーションとハードウェア資源との間に介在して、ユーザサービスを提供する際に、アプリケーションの少なくとも2つが共通的に必要とするハードウェア資源の管理、実行制御並びに画像形成処理を行う各種コントロールサービスからなるプラットフォームを備えた画像形成装置（複合機）を発明した。

【0 0 0 5】

このような新規な複合機では、アプリケーションの少なくとも2つが共通的に必要とするサービスを提供するコントロールサービスをアプリケーションと別個に設けた構成となっているため、複合機の出荷後にユーザもしくは第三者であるサードベンダが新規なアプリケーションを開発して、複合機に搭載可能な構成となっている。このため、かかる複合機では、出荷時に搭載されているコピー、プリンタ、スキャナ、ファクシミリといった既存のアプリケーション以外に、従来の複合機とは異なるユーザやサードベンダが開発した新規アプリケーションなどを複合機にコピーした後、そのアプリケーションを起動可能な状態にするためのインストール処理をする必要がある。上記した複合機の既存のアプリケーションの場合は、バージョンの整合性や使用リソース量があらかじめわかっているため、インストールして起動させたとしても、バージョンの不整合やリソース不足による動作の不具合が生じるおそれはない。しかしながら、サードベンダなどが開発する新規アプリ

ケーションの場合は、搭載する複合機の状況（バージョンやリソース容量）まで考慮して開発していないため、その新規アプリケーションを起動させた場合に複合機の動作が不安定になる可能性がある。そこで、とりあえずインストールしておいて、起動までにチェックすれば動作が不安定になることが防止できるとも考えられる。しかし、起動させると複合機の動作が不安定になる可能性のあるアプリケーションをインストールしておくのは無意味なことである。このように、新規アプリケーションが搭載可能な新規な複合機では、新規アプリケーションのバージョンチェックやリソースチェックを行わずに起動させると、動作が不安定となったり、動作が途中で停止するなど従来の複合機では問題にならなかった新規な課題が生じてくる。

#### 【0006】

この発明は上記に鑑みてなされたもので、新規アプリケーションを複合機に搭載して、起動可能な状態とするインストール処理の前に、インストールが可能なアプリケーションか否かを判断するために必要なアプリケーションに関する正確なアプリ情報を取得できる画像形成装置およびアプリ情報取得方法を得ることを目的とする。

#### 【課題を解決するための手段】

#### 【0007】

上記目的を達成するため、請求項1にかかる発明は、アプリケーションをインストール可能に構成された画像形成装置であって、アプリケーションを前記画像形成装置にインストールすることが可能か否かを判断するために使用されるアプリケーション情報の取得を行うアプリケーション情報取得手段を備えたことを特徴とする画像形成装置である。

#### 【0008】

本発明によれば、インストールが可能なアプリケーションか否かを正確な判断を下すことができる。

#### 【0009】

また、請求項2にかかる発明は、請求項1に記載の画像形成装置において、前記アプリケーション情報取得手段は、前記画像形成装置におけるアプリケーションが自ら保有するアプリケーション情報を取得することを特徴とする。

#### 【0010】

本発明によれば、アプリケーション情報取得手段は、アプリケーションから直接正確な情報を取得することができる。

#### 【0011】

請求項3にかかる発明は、請求項2の記載において、前記アプリケーションが自ら保有するアプリケーション情報は、当該アプリケーションの実行ファイル中に含まれる情報であることを特徴とする。これにより、アプリケーションから直接正確な情報を取得することができる。

#### 【0012】

また、請求項4にかかる発明は、請求項2の記載において、前記アプリケーション情報取得手段は、前記画像形成装置におけるアプリケーションとは別のファイルから前記アプリケーション情報を取得することを特徴とするものである。

#### 【0013】

本発明により、例えば、ネットワーク経由でアプリケーション情報を取得することが可能となる。

#### 【0014】

また、請求項5にかかる発明は、請求項3に記載の画像形成装置において、前記アプリケーション情報取得手段は、前記アプリケーションを仮起動させることにより、前記アプリケーション情報を取得することを特徴とする。

#### 【0015】

本発明によれば、通常起動前のアプリケーションに対しても、正確なアプリ情報を取得することができる。

#### 【0016】

また、請求項 6 にかかる発明は、請求項 5 に記載の画像形成装置において、前記アプリケーション情報取得手段は、仮起動させたアプリケーションからプロセス間通信により前記アプリケーション情報を取得することを特徴とする。

【 0 0 1 7 】

本発明により、通常起動前のアプリケーションに対してプロセス間通信により、正確なアプリ情報を取得することができる。

【 0 0 1 8 】

請求項 7 にかかる発明は、請求項 5 に記載の画像形成装置において、前記アプリケーションは、前記アプリケーション情報取得手段からの要求に基づき、仮起動するか通常起動するかを判断するステップと、仮起動の場合に、前記アプリケーション情報取得手段と通信することにより、アプリケーション情報を前記アプリケーション情報取得手段に提供するステップを前記画像形成装置に実行させることを特徴とする。

【 0 0 1 9 】

本発明によれば、アプリケーションが、アプリケーション情報取得手段からの要求に基づき仮起動するか通常起動するかを判断する手段を備えるので、仮起動するか通常起動するかを選択することができる。

【 0 0 2 0 】

また、請求項 8 にかかる発明は、請求項 1 に記載の画像形成装置において、前記アプリ情報取得手段が取得したアプリケーション情報を記憶する記憶手段をさらに備えたことを特徴とする。

【 0 0 2 1 】

本発明によれば、アプリ情報取得手段が取得したアプリ情報を記憶手段に記憶するようにしたため、一旦取得したアプリ情報を必要に応じていつでも取り出せることから、データベースのように使用することが可能となり、アプリ情報取得に要する時間を短縮して、処理の迅速化が図れる。

【 0 0 2 2 】

また、請求項 9 にかかる発明は、請求項 8 に記載の画像形成装置において、前記記憶手段は、ハードディスク、不揮発性メモリ、および、SD カード等のメモリカードの少なくとも一つであって、前記アプリ情報取得手段は、それらの記憶媒体に記憶された情報を読み出すことによりアプリ情報を取得することを特徴とする。

【 0 0 2 3 】

本発明によれば、種々の記憶媒体をアプリ情報の記憶手段として利用することが可能となり、状況に応じて記憶媒体を使い分けることができる。

【 0 0 2 4 】

また、請求項 1 0 にかかる発明は、請求項 1 ～ 9 のいずれか一つに記載の画像形成装置において、前記アプリケーション情報取得手段が取得するアプリケーション情報は、アプリケーションに固有のプロダクト ID を含むことを特徴とする。

【 0 0 2 5 】

本発明によれば、アプリケーション情報のやり取りをこのプロダクト ID に基づいて、簡易かつ迅速に行うことができるとともに、アプリ情報が一元管理できることから、情報の信頼性を向上させることができる。

【 0 0 2 6 】

また、請求項 1 1 にかかる発明は、請求項 1 0 に記載の画像形成装置において、前記アプリケーション情報は、前記プロダクト ID に加えて、対応するアプリケーションのベンダー名とアプリケーション名、バージョン情報と使用リソース情報の少なくとも一方を含むことを特徴とする。

【 0 0 2 7 】

本発明によれば、アプリケーションの起動時に問題となるバージョンの整合性やリソース不足による動作の不安定要因の有無を正確に判断することができる。

【 0 0 2 8 】



また、請求項 1 2 にかかる発明は、請求項 1 0 または 1 1 に記載の画像形成装置において、前記記憶手段は、インストールを可能なアプリケーションのプロダクト I D を記憶し、判断対象となるアプリケーションのプロダクト I D がその記憶手段に記憶されているか否かによって、インストール可能なアプリケーションか否かを自動的に判断することを特徴とする。

#### 【 0 0 2 9 】

本発明によれば、簡易な構成により迅速かつ正確にインストール可能なアプリケーションか否かを自動判断することができる。

#### 【 0 0 3 0 】

また、請求項 1 3、1 4 に記載の発明は、画像形成装置が仮想アプリケーションサービスを有し、仮想アプリケーションサービスが前記アプリケーション情報取得手段を含むとしたものである。

#### 【 0 0 3 1 】

また、請求項 1 5 ～ 2 6 に記載の発明は、アプリケーション情報取得のための方法の発明であり、上記画像形成装置と同様の作用効果を奏する。また、請求項 2 7 ～ 2 8 に記載の発明は、アプリケーション情報取得のためのプログラムの発明であり、上記画像形成装置と同様の作用効果を奏する。

#### 【発明の効果】

#### 【 0 0 3 2 】

以上説明したように、本発明によれば、アプリケーション情報を画像形成装置が取得するので、そのアプリケーション情報を用いて、インストールが可能なアプリケーションか否かを判断することができる。

#### 【発明を実施するための最良の形態】

#### 【 0 0 3 3 】

以下に添付図面を参照して、この発明にかかる画像形成装置およびアプリ情報取得方法の好適な実施の形態を詳細に説明する。

#### 【 0 0 3 4 】

図 1 は、この発明の本実施の形態である画像形成装置（以下、「複合機」という）の構成を示すブロック図である。図 1 に示すように、複合機 1 0 0 は、白黒レーザプリンタ（B&W LP） 1 0 1 と、カラーレーザプリンタ（Color LP） 1 0 2 と、スキャナ、ファクシミリ、ハードディスク、メモリ、ネットワークインタフェースなどのハードウェアリソース 1 0 3 を有するとともに、プラットフォーム 1 2 0 とアプリケーション（以下、アプリともいう） 1 3 0 と仮想アプリケーションサービス（V A S : V i r t u a l A p p l i c a t i o n S e r v i c e） 1 4 0 から構成されるソフトウェア群 1 1 0 とを備えている。

#### 【 0 0 3 5 】

また、本実施の形態では、アプリ 1 3 0 とプラットフォーム 1 2 0 との間に仮想アプリケーションサービス（V A S） 1 4 0 を配置している。この仮想アプリケーションサービス（V A S） 1 4 0 は、コントロールサービスをサーバとしたクライアントプロセスとして動作し、かつアプリケーションをクライアントとしたサーバプロセスとして動作するもので、アプリ情報取得スレッドがこの仮想アプリケーションサービス 1 4 0 に含まれている。この V A S 1 4 0 は、あるアプリを起動可能な状態とするインストール処理を行う前に、アプリ情報を取得して、そのアプリ情報ファイルを記憶手段としてのハードディスク（H D） 2 0 0 に生成する。取得するアプリ情報の例としては、ここでは、プロダクト情報を取得するようにして、プロダクト I D、ベンダー名、アプリ名、バージョン、連絡先（電話番号）、必要リソース、インストール時判定、関連アプリ情報などが含まれている。このプロダクト情報は、アプリの実行ファイル中に持っていたり、ファイルとしてネットワーク経由で取得したり、あるいは、以前に取得したプロダクト情報をハードディスク 2 0 0、不揮発性メモリである N V R A M 1 0 4、アプリ専用のアプリ情報テーブル 1 0 6 などに格納されている場合は、そのプロダクト情報を読み出すことによっても利用す

ることができる。

#### 【 0 0 3 6 】

アプリの実行ファイル中にあるアプリ情報は、V A S 1 4 0 がアプリを仮起動することにより、V A S 1 4 0 とアプリの間のプロセス間通信により、V A S 1 4 0 が取得することが可能である。

#### 【 0 0 3 7 】

また、アプリ自身は複合機 1 0 0 のハードディスク 2 0 0 にコピーしたり、P C カード、I C カード、S D カードといった種々のデータカードを介してコピーされた後、起動できるような状態とするインストール処理が行われることになる。本実施の形態では、インストール処理を行う前に、アプリに関する情報であるアプリ情報を取得し、このアプリ情報に基づいてインストール処理が可能なアプリか否かを判断するようにする。このインストール処理が可能な否かの判断は、インストール処理後、そのアプリを起動させたとした場合に、インストールした複合機の既存のアプリや他のインストール済みの新規アプリとの関係において、リソース不足やバージョンの整合性等の問題で複合機自体の動作が不安定となったり、動作が停止するおそれがあるか否かによる。このような判断を行う理由は、起動できないアプリをインストールしても意味がない上、インストールされていると、誤って起動させた場合に複合機自体の動作に支障が出るからである。

#### 【 0 0 3 8 】

プラットフォーム 1 2 0 は、アプリケーションからの処理要求を解釈してハードウェア資源の獲得要求を発生させるコントロールサービスと、一または複数のハードウェア資源の管理を行い、コントロールサービスからの獲得要求を調停するシステムリソースマネージャ (SRM) 1 2 3 と、汎用 O S 1 2 1 とを有している。

#### 【 0 0 3 9 】

コントロールサービスは、複数のサービスモジュールから形成され、S C S (システムコントロールサービス) 1 2 2 と、E C S (エンジンコントロールサービス) 1 2 4 と、M C S (メモリコントロールサービス) 1 2 5 と、O C S (オペレーションパネルコントロールサービス) 1 2 6 と、F C S (ファックスコントロールサービス) 1 2 7 と、N C S (ネットワークコントロールサービス) 1 2 8 とから構成されている。なお、このプラットフォーム 1 2 0 は、あらかじめ定義された関数により前記アプリケーション 1 3 0 から処理要求を受信可能とするアプリケーションプログラムインタフェース (API) を有している。

#### 【 0 0 4 0 】

汎用 O S 1 2 1 は、U N I X (登録商標) などの汎用オペレーティングシステムであり、プラットフォーム 1 2 0 並びにアプリケーション 1 3 0 の各ソフトウェアをそれぞれプロセスとして並列実行する。

#### 【 0 0 4 1 】

SRM 1 2 3 のプロセスは、S C S 1 2 2 とともにシステムの制御およびリソースの管理を行うものである。SRM 1 2 3 のプロセスは、スキャナ部やプリンタ部などのエンジン、メモリ、HDD ファイル、ホスト I / O (セントロ I / F、ネットワーク I / F、I E E E 1 3 9 4 I / F、R S 2 3 2 C I / F など) のハードウェア資源を利用する上位層からの要求にしたがって調停を行い、実行制御する。

#### 【 0 0 4 2 】

具体的には、この SRM 1 2 3 は、要求されたハードウェア資源が利用可能であるか (他の要求により利用されていないかどうか) を判断し、利用可能であれば要求されたハードウェア資源が利用可能である旨を上位層に伝える。また、SRM 1 2 3 は、上位層からの要求に対してハードウェア資源の利用スケジューリングを行い、要求内容 (例えば、プリンタエンジンにより紙搬送と作像動作、メモリ確保、ファイル生成など) を直接実施している。

#### 【 0 0 4 3 】

S C S 1 2 2 のプロセスは、アプリ管理、操作部制御、システム画面表示、L E D 表示

、リソース管理、割り込みアプリ制御などを行う。

【0044】

ECS124のプロセスは、白黒レーザプリンタ（B&W LP）101、カラーレーザプリンタ（Color LP）102、スキャナ、ファクシミリなどからなるハードウェアリソースのエンジンの制御が行われる。

【0045】

MCS125のプロセスは、画像メモリの取得および解放、ハードディスク装置（HDD）の利用、画像データの圧縮および伸張などを行う。

【0046】

FCS127のプロセスは、システムコントローラの各アプリ層からPSTN/I SDN網を利用したファクシミリ送受信、BKM（バックアップSRAM）で管理されている各種ファクシミリデータの登録／引用、ファクシミリ読みとり、ファクシミリ受信印刷、融合送受信を行うためのAPIを提供する。

【0047】

NCS128のプロセスは、ネットワークI/Oを必要とするアプリケーションに対して共通に利用できるサービスを提供するためのプロセスであり、ネットワーク側から各プロトコルによって受信したデータを各アプリケーションに振り分けたり、アプリケーションからデータをネットワーク側に送信する際の仲介を行う。具体的には、ftpd、httpd、lpd、snmpd、telnetd、smtpdなどのサーバデーモンや、同プロトコルのクライアント機能などを有している。

【0048】

OCS126のプロセスは、オペレータ（ユーザ）と本体制御間の情報伝達手段となるオペレーションパネル（操作パネル）105の制御を行う。OCS126は、オペレーションパネル105からキー押下をキーイベントとして取得し、取得したキーに対応したキーイベント関数をSCS122に送信するOCSプロセスの部分と、アプリケーション130またはコントロールサービスからの要求によりオペレーションパネルに各種画面を描画出力する描画関数やその他オペレーションパネルに対する制御を行う関数などがあらかじめ登録されたOCSライブラリの部分とから構成される。このOCSライブラリは、アプリケーション130およびコントロールサービスの各モジュールにリンクされて実装されている。なお、OCS126のすべてをプロセスとして動作させるように構成しても良く、あるいはOCS126のすべてをOCSライブラリとして構成しても良い。

【0049】

アプリケーション130は、ページ記述言語（PDL）、PCLおよびポストスクリプト（PS）を有するプリンタ用のアプリケーションであるプリンタアプリ111と、コピー用アプリケーションであるコピーアプリ112と、ファクシミリ用アプリケーションであるファックスアプリ113と、スキャナ用アプリケーションであるスキャナアプリ114と、ネットワークファイル用アプリケーションであるネットファイルアプリ115と、工程検査用アプリケーションである工程検査アプリ116とを有している。これらの各アプリは、複合機特有（画像形成装置特有）のアプリであり、起動時にVAS140に対して自プロセスのプロセスIDとともにアプリ登録要求メッセージを送信し、アプリ登録要求メッセージを受信したVAS140によって、起動したアプリに対する登録処理が行われるようになっていく。

【0050】

アプリケーション130の各プロセス、コントロールサービスの各プロセスは、関数呼び出しとその戻り値送信およびメッセージの送受信によってプロセス間通信を行いながら、コピー、プリンタ、スキャナ、ファクシミリなどの画像形成処理にかかるユーザサービスを実現している。

【0051】

このように、本実施の形態にかかる複合機100には、複数のアプリケーション130および複数のコントロールサービスが存在し、いずれもプロセスとして動作している。そ

して、これらの各プロセス内部には、一または複数のスレッドが生成されて、スレッド単位の並列実行が行われる。そして、コントロールサービスがアプリケーション 1 3 0 に対し共通サービスを提供しており、このため、これらの多数のプロセスが並列動作、およびスレッドの並列動作を行って互いにプロセス間通信を行って協調動作をしながら、コピー、プリンタ、スキャナ、ファクシミリなどの画像形成処理にかかるユーザサービスを提供するようになっている。また、複合機 1 0 0 には、サードベンダなどの第三者がコントロールサービス層の上のアプリケーション層に新規アプリ 1 1 7, 1 1 8 を開発して搭載することが可能となっている。図 1 では、この新規アプリ 1 1 7, 1 1 8 を搭載した例を示している。

#### 【0052】

なお、本実施の形態にかかる複合機 1 0 0 では、複数のアプリケーション 1 3 0 のプロセスと複数のコントロールサービスのプロセスとが動作しているが、アプリケーション 1 3 0 とコントロールサービスのプロセスがそれぞれ単一の構成とすることも可能である。また、各アプリケーション 1 3 0 は、アプリケーションごとに追加または削除することができる。

#### 【0053】

図 2 に複合機 1 0 0 のハードウェア構成例を示す。

#### 【0054】

複合機 1 0 0 は、コントローラ 1 6 0 と、オペレーションパネル 1 7 5 と、ファックスコントロールユニット (FCU) 1 7 6 と、プリンタ等の画像形成処理に特有のハードウェア資源であるエンジン部 1 7 7 とを含む。コントローラ 1 6 0 は、CPU 1 6 1 と、システムメモリ 1 6 2 と、ノースブリッジ (NB) 1 6 3 と、サウスブリッジ (SB) 1 6 4 と、ASIC 1 6 6 と、ローカルメモリ 1 6 7 と、HDD 1 6 8 と、ネットワークインターフェースカード (NIC) 1 6 9 と、SDカード用スロット 1 7 0 と、USBデバイス 1 7 1 と、IEEE 1 3 9 4 デバイス 1 7 2 と、センタロニクス 1 7 3 とを含む。なお、メモリ 1 6 2、1 6 7 は RAM、ROM 等を含む。FCU 1 7 6 およびエンジン部 1 7 7 は、コントローラ 1 6 0 の ASIC 1 6 6 に PCI バス 1 7 8 で接続されている。

#### 【0055】

CPU 1 6 1 が、複合機 1 0 0 にインストールされるアプリケーション、コントロールサービス等のプログラムを、メモリから読み出して実行する。

#### 【0056】

図 3 は、本実施の形態にかかる複合機 1 0 0 の VAS 1 4 0 の構成と、VAS 1 4 0 と各アプリ、コントロールサービス層 1 5 0 および汎用 OS 1 2 1 との関係を示すブロック図である。なお、図 3 では、アプリケーション 1 3 0 の例として、プリンタアプリ 1 1 1、コピーアプリ 1 1 2、新規アプリ 1 1 7, 1 1 8 を示しているが、他のアプリでも同様の構成である。

#### 【0057】

仮想アプリケーションサービス (VAS) 1 4 0 のプロセスには、ディスパッチャ 1 4 5 と、制御スレッド 1 4 4 と、アプリ情報取得手段としてのアプリ情報取得スレッド 1 4 1 と、オペレーションパネルに対して操作画面を生成する画面表示スレッド 1 4 2、および、取得したアプリ情報に基づいてアプリのインストールが可能か否かをチェックするアプリチェックスレッド 1 4 3 が動作している。

#### 【0058】

ディスパッチャ 1 4 5 は、アプリケーション 1 3 0 やコントロールサービスからのメッセージ受信を監視し、受信したメッセージに応じて制御スレッド 1 4 4、アプリ情報取得スレッド 1 4 1、画面表示スレッド 1 4 2、アプリチェックスレッド 1 4 3 に対して処理要求を行うものである。本実施の形態の複合機 1 0 0 では、ディスパッチャ 1 4 5 は、コントロールサービスからインストールされているアプリに対するアプリ起動要求メッセージを受信したとき、受信したアプリ起動要求メッセージを制御スレッド 1 4 4 に送信し、制御スレッド 1 4 4 からアプリ情報取得スレッド 1 4 1、画面表示スレッド 1 4 2、アプ

リチェックスレッド 1 4 3 に対して処理要求を行うようになっている。

#### 【0059】

制御スレッド 1 4 4 は、ディスパッチャ 1 4 4 からのアプリ起動要求メッセージを受信して、アプリ情報取得処理、画面表示処理、およびアプリチェック処理を行う。ここで、アプリチェック処理とは、アプリをインストール後に起動させた場合に複合機に対して影響を与える可能性が有るか無いかをチェックする処理のことである。例えば、インストール要求のあったアプリと、他のインストール済みのアプリのアプリ情報を取得して、バージョンチェックやリソースチェックを行うことによって、インストール後に起動させても動作に影響を与えないか否かを判断することが可能となる。その際、画面表示スレッド 1 4 2 は、ハードディスク (H D) 2 0 0 中のアプリ情報ファイル 2 0 1 に格納したアプリ情報を用いてオペレーションパネル 1 0 5 上に各アプリのアプリ情報を表示するので、インストールしようとしているアプリの状況を把握することができる。従って、アプリチェックスレッド 1 4 3 によってインストールが不可能であると判断されても、どのような理由でインストールできなかったのかを理解することができる。

#### 【0060】

また、制御スレッド 1 4 3 は、アプリチェックスレッド 1 4 3 によってインストールが可能と判断されたり、あるいは、後述するように、インストールされているアプリが不要になった場合などに削除するアンインストール処理などを制御する。

#### 【0061】

アプリ情報取得スレッド 1 4 1 は、制御スレッド 1 4 3 からの処理要求を受けると、インストール前のアプリに対しては、仮起動を行うことによってプロセス間通信が可能な状態として、アプリ情報を取得する。また、インストールされているアプリは、起動させることで同様にプロセス間通信によるアプリ情報取得が可能となる。このようにして、取得された各アプリのアプリ情報は、アプリ情報ファイル 2 0 1 としてハードディスク (H D) 2 0 0 に生成される。かかるアプリ情報は、アプリごとのレコードとして記録される。

#### 【0062】

上記 V A S 1 4 0 のプログラムは、ソフトウェア開発キット (S D K : S o f t w a r e D e v e l o p m e n t K i t) 等の一部または全部として、C D - R O M または F D (フレキシブルディスク) などの記憶媒体に実行可能な形式またはインストール可能な形式のファイルで提供される。また、このような実行可能な形式またはインストール可能な形式の V A S 1 4 0 のプログラムファイルを、ネットワーク経由で取得可能な方法で提供するようにしても良い。

#### 【0063】

図 4 は、V A S 1 4 0 とアプリ情報を格納するハードディスク 2 0 0、アプリ情報テーブル 1 0 6、N V R A M 1 0 4 との関係を示す説明図である。図 4 に示すように、ハードディスク 2 0 0 に格納されているプロダクト情報は、各アプリ単位ごとに管理され、プロダクト I D が特定できれば、これに対応するアプリ情報も特定することができる。また、このハードディスク 2 0 0 には、複合機にコピーされたインストール前のアプリ A やアプリ C のデータが格納されており、また、アプリ B は、I C カードなどに格納されたデータを使ってインストール処理を行うものである。図 4 中の破線で示した F は、特定のパスにある実行ファイル群を示している。

#### 【0064】

N V R A M 1 0 4 は、電源を切ってもメモリ内容を保持する不揮発性メモリであり、アプリ A、アプリ B のように、各アプリごとにプロダクト I D、起動情報、付加情報がそれぞれ格納されている。N V R A M 1 0 4 には、アプリのインストール処理を行うことによって情報が記録される。また、電源 O N 時に当該アプリが起動されるか否かを示す情報が起動情報として設定される。

#### 【0065】

また、アプリ情報テーブルは、アプリに関する情報を各アプリごとにテーブルデータとして格納したものである。このため、アプリ A のプロダクト I D を入力すれば、このアプ

り A のアプリ名、バージョン、インストール属性等のアプリ情報を取り出すことができる。そして、オペレーションパネルへの表示等を行うことができる。

#### 【0066】

図 5 は、本実施の形態の複合機においてアプリに対してインストール処理かアンインストール処理を行う場合の全体フローチャートであり、図 6 は、図 5 のステップ S 4 0 1 におけるアプリ情報取得処理のサブルーチンを示すフローチャートである。

#### 【0067】

まず、図 5 のステップ S 4 0 1 では、ディスパッチャ 1 4 5 が制御スレッド 1 4 4 を介してアプリ情報取得スレッド 1 4 1 に対して、インストール前、あるいは、インストール済みのアプリに対するアプリ情報取得処理を要求する。アプリ情報スレッド 1 4 1 は、後述する図 6 におけるアプリ情報取得処理を行って、アプリ情報ファイル 2 0 1 を生成する。

#### 【0068】

また、上記ステップ S 4 0 1 では、生成されたアプリ情報ファイル 2 0 1 から制御スレッド 1 4 4 がアプリ情報を読み出して、オペレーションパネル 1 0 5 に表示する操作画面を生成して、画面表示が行われる（ステップ S 4 0 2）。この表示画面は、コピーされたアプリに関するアプリ名、バージョン、リソース等の情報を併記して表示するとともに、選択画面に沿ってユーザが選択していただくだけで、インストールの可否やインストール不可能になった場合の状況等が画面表示により理解できるようになっている。ユーザは、その画面表示に従って選択処理を行う。

#### 【0069】

全ての項目についてユーザの選択処理が行われた後は、ステップ S 4 0 3 において、ユーザ操作待機の後（ステップ S 4 0 3）、ユーザ要求の有無が判断される（ステップ S 4 0 4）。ここでは、ユーザの要求としては、アプリのインストール要求か、アンインストール要求のいずれかである。ユーザ要求がインストールで、許可されるならば、図 4 に示す NVRAM 1 0 4 に対して登録処理が行われる（ステップ S 4 0 5）。また、ユーザ要求がアンインストールであるならば、図 4 に示す NVRAM 1 0 4 からアプリ情報を削除することによってアンインストールされることになる（ステップ S 4 0 6）。

#### 【0070】

図 6 は、図 5 におけるアプリ情報取得処理動作を詳細に説明したものである。インストール前あるいは、起動前のアプリに対してはそのままではアプリ情報を取得することができないため、VAS 1 4 0 は、情報取得対象のアプリを仮起動させ（ステップ S 5 0 1）、プロセス間通信により、所望の情報を要求することでアプリ情報を取得することができる。本実施の形態では、アプリ情報の一元管理が可能なプロダクト ID を要求してこれを取得すると（ステップ S 5 0 2）、NVRAM 1 0 4 を参照することにより、そのアプリが登録済みのアプリか否かがわかる（ステップ S 5 0 3）。

#### 【0071】

NVRAM 1 0 4 に登録済みでない場合は、プロダクト情報ファイルの有無が判断される（ステップ S 5 0 4）、プロダクト情報がない場合は、アプリよりプロセス間通信を使って各種情報を取得するようにする（ステップ S 5 0 5）。これにより、プロダクト情報ファイルを生成した後（ステップ S 5 0 6）、インストール判定処理が行われる（ステップ S 5 0 7）。そして、その判定処理結果をプロダクト情報に反映させるようにする（ステップ S 5 0 8）。インストール判定は、例えば、アプリから取得した使用リソース情報と、システムが有するリソース情報を比較することにより行う。

#### 【0072】

そして、ステップ S 5 0 9 では、その判定結果に基づいて、インストールの可否が判断される。また、上記ステップ S 5 0 4 で、プロダクト情報ファイルが有る場合は、プロダクト情報よりアプリ情報取得がなされ（ステップ S 5 1 0）、インストール時の判定情報取得が行われた後（ステップ S 5 1 1）、上記したステップ S 5 0 9 において、インストールの可否が判断される。

**【0073】**

ステップS509において、インストール可能であれば、インストール可能フラグの設定を行い（ステップS512）、アプリ情報テーブルへの書き込みが行われる（ステップS516）。また、インストールが不可の場合は、インストール不可フラグの設定が行われて（ステップS513）、アプリ情報テーブルへの書き込みが行われる（ステップS516）。さらに、上記したステップS503において、NVRAM104にプロダクトIDが登録されているアプリの場合は、NVRAM104よりアプリ情報取得が行われ（ステップS514）、インストール済みのフラグの設定が行われて（ステップS515）、アプリ情報テーブルへの書き込みが行われる（ステップS516）。

**【0074】**

そして、全アプリに対する処理が終了していなければ、再び上記ステップS501に戻って、全アプリに対する処理が終了するまで、上記処理が繰り返される。

**【0075】**

このように、本実施の形態の複合機100では、新規アプリをインストール処理する場合に、仮想アプリケーションサービス140のアプリ情報取得スレッド141を使って、全アプリに対してアプリ情報の取得が行われるため、この取得したアプリ情報に基づいて、インストールした後に当該アプリを起動した場合の動作状況を正確に把握することが可能となり、インストール後に起動させた場合でも、複合機の動作が不安定になったり、動作が停止するような状況を防止することが可能となる。

**【0076】**

次に、アプリケーションの仮起動についてより詳しく説明する。仮起動は、複合機のリソースを使用することになるアプリの通常起動（アプリ本来の機能を奏するための起動を通常起動と呼ぶ）とは別の起動である。仮起動では、アプリはアプリ本来の動作に必要なメモリ確保等のリソース取得を行わず、VASとのプロセス間通信処理のみを行う。そして、アプリは、VASに情報を提供する。仮起動したアプリのプロセスは、VASとの通信処理が終了すれば終了する。また、アプリの仮起動に関する機能は、アプリ本来の機能によらず、本実施の形態における複合機で動作するアプリに共通する機能である。従って、例えば、ベンダーがアプリを開発する場合、ベンダーに、仮起動の機能を含むプログラムテンプレートを提供し、そのプログラムテンプレートを用いてベンダーが複合機用のアプリを開発することができる。なお、上記のアプリ情報は、例えば、ベンダーが、アプリ情報を記録したインクルードファイルを作成し、アプリのコンパイル時にインクルードすることにより、アプリの実行ファイル中に含めることができる。

**【0077】**

アプリが仮起動の機能を持つことにより、通常起動をすることなくVASとの通信によりVASにアプリ情報を提供でき、VASがインストール可否のチェックを行うことができる。

**【0078】**

図7に、仮起動の機能を含むアプリのプログラム記述（メイン関数）の概要を示す。なお、この記述を上記のプログラムテンプレートとして提供する。

**【0079】**

図7に示すように、このプログラム記述は、アプリケーションを仮起動するか通常起動するかを引数（-v）によって指定する。これにより、VASがアプリを起動する際に、通常起動と仮起動とを容易に使い分けることができる。すなわち、引数（-v）を使って仮起動を指定すると、仮起動が実行され、アプリ情報提供処理がなされる。また、仮起動が指定されていない場合は、通常起動を行って、アプリ本来の動作を行う。

**【0080】**

なお、上記の実施の形態では、VASがアプリから情報を取得してインストールの可否を判断し、アプリの登録を行っていたが、図8に示すシーケンスのように、インストール可であるとVASが判断した後に、VASからアプリに対して、アプリのインストールを開始してよいか否かの問い合わせをし（ステップS601）、アプリがその問い合わせに

対してインストール可の通知を V A S に返した場合に（ステップ S 6 0 2）、V A S がアプリの登録をするようにしてもよい（ステップ S 6 0 3）。

【0 0 8 1】

なお、ステップ S 6 0 1 とステップ S 6 0 2 の間で、アプリが V A S からシステム情報（リソース情報あるいは機器構成情報等）を取得し、その情報に基づき、インストールできるか否かをアプリが判断するようにしてもよい。

【0 0 8 2】

なお、V A S がアプリ情報を取得する方法は、アプリの実行ファイル中に含められたアプリ情報を取得する方法以外にもある。その方法は、例えば、アプリ情報をアプリとは別のファイルに作成しておき、そのファイルの内容を V A S が読み出すというものである。そのファイルを複合機の外部のサーバに備えて置くことにより、ネットワーク経由でアプリ情報を取得することもできる。

【0 0 8 3】

なお、本発明は、上記の実施の形態に限定されることなく、特許請求の範囲内において、種々変更・応用が可能である。

【図面の簡単な説明】

【0 0 8 4】

【図 1】 実施の形態 1 にかかる複合機の構成を示すブロック図である。

【図 2】 実施の形態 1 にかかる複合機のハードウェア構成を示すブロック図である。

【図 3】 本実施の形態にかかる複合機の V A S の構成と、V A S と各アプリ、コントロールサービス層および汎用 O S との関係を示すブロック図である。

【図 4】 V A S とアプリ情報を格納するハードディスク、アプリ情報テーブル、N V R A M との関係を示す説明図である。

【図 5】 本実施の形態の複合機においてアプリに対してインストール処理かアンインストール処理を行う場合の全体フローチャートである。

【図 6】 図 5 のステップ S 4 0 1 におけるアプリ情報取得処理のサブルーチンを示すフローチャートである。

【図 7】 仮起動の機能を含むアプリのプログラム記述（メイン関数）の概要を示す図である。

【図 8】 V A S とアプリのネゴシエーションを示す図である。

【符号の説明】

【0 0 8 5】

- 1 0 0 複合機
- 1 0 1 白黒レーザプリンタ
- 1 0 2 カラーレーザプリンタ
- 1 0 4 N V R A M
- 1 0 5 オペレーションパネル
- 1 1 0 ソフトウェア群
- 1 1 1 プリンタアプリ
- 1 1 2 コピーアプリ
- 1 1 3 ファックスアプリ
- 1 1 4 スキャナアプリ
- 1 1 5 ネットファイルアプリ
- 1 1 6 工程検査アプリ
- 1 1 7, 1 1 8 新規アプリ
- 1 2 0 プラットホーム
- 1 2 1 汎用 O S
- 1 2 2 S C S
- 1 2 3 S R M
- 1 2 4 E C S

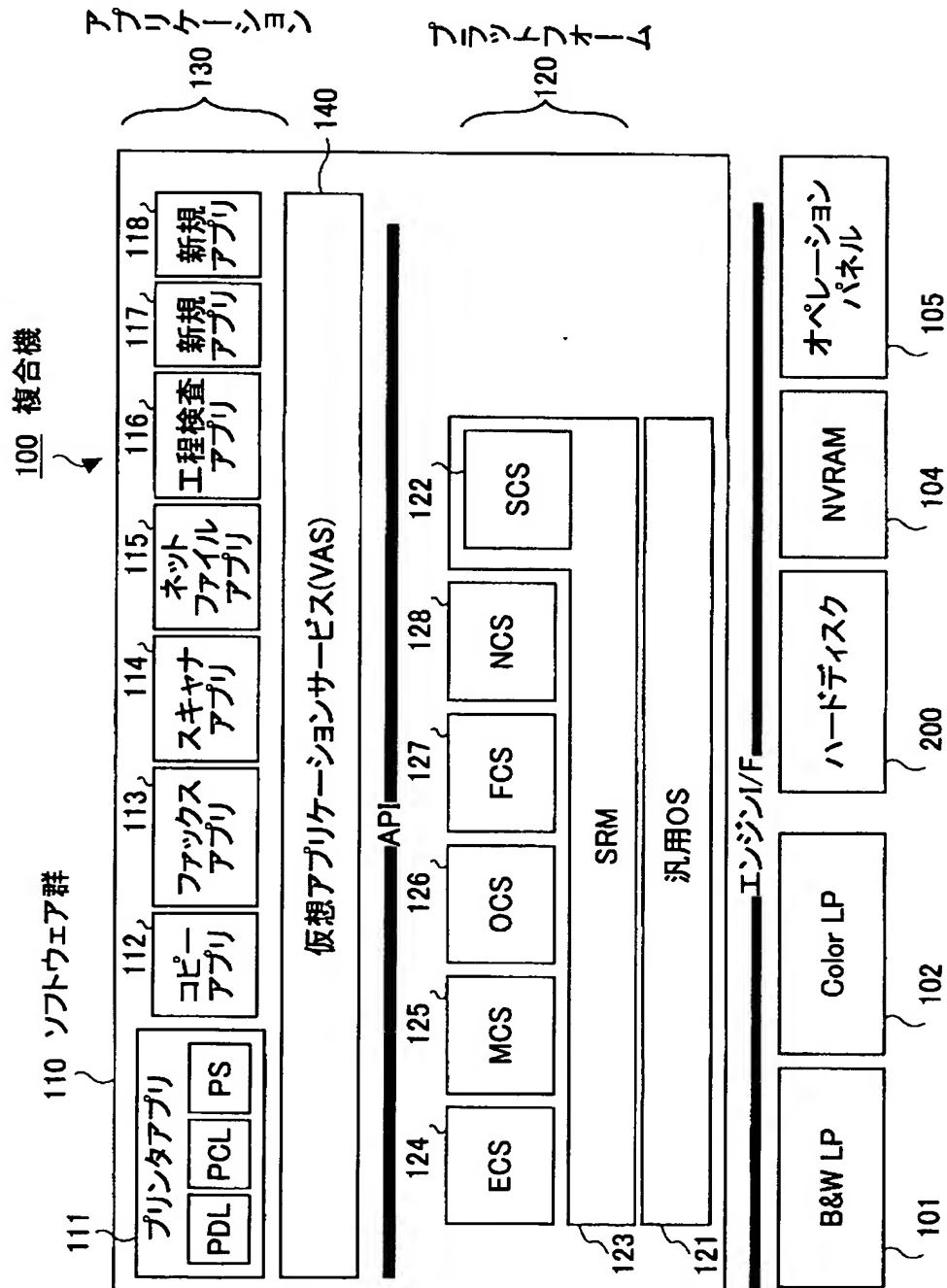


1 2 5 M C S  
 1 2 6 O C S  
 1 2 7 F C S  
 1 2 8 N C S  
 1 3 0 アプリケーション  
 1 4 0, 8 4 1 ~ 8 4 8 仮想アプリケーションサービス (V A S)  
 1 4 1 アプリ情報取得スレッド  
 1 4 2 画面表示スレッド  
 1 4 3 アプリチェックスレッド  
 1 4 4 制御スレッド  
 1 4 5 ディスパッチャ  
 1 5 0 コントロールサービス層  
 2 0 0 ハードディスク (H D)  
 2 0 1 アプリ情報ファイル

【書類名】 図面

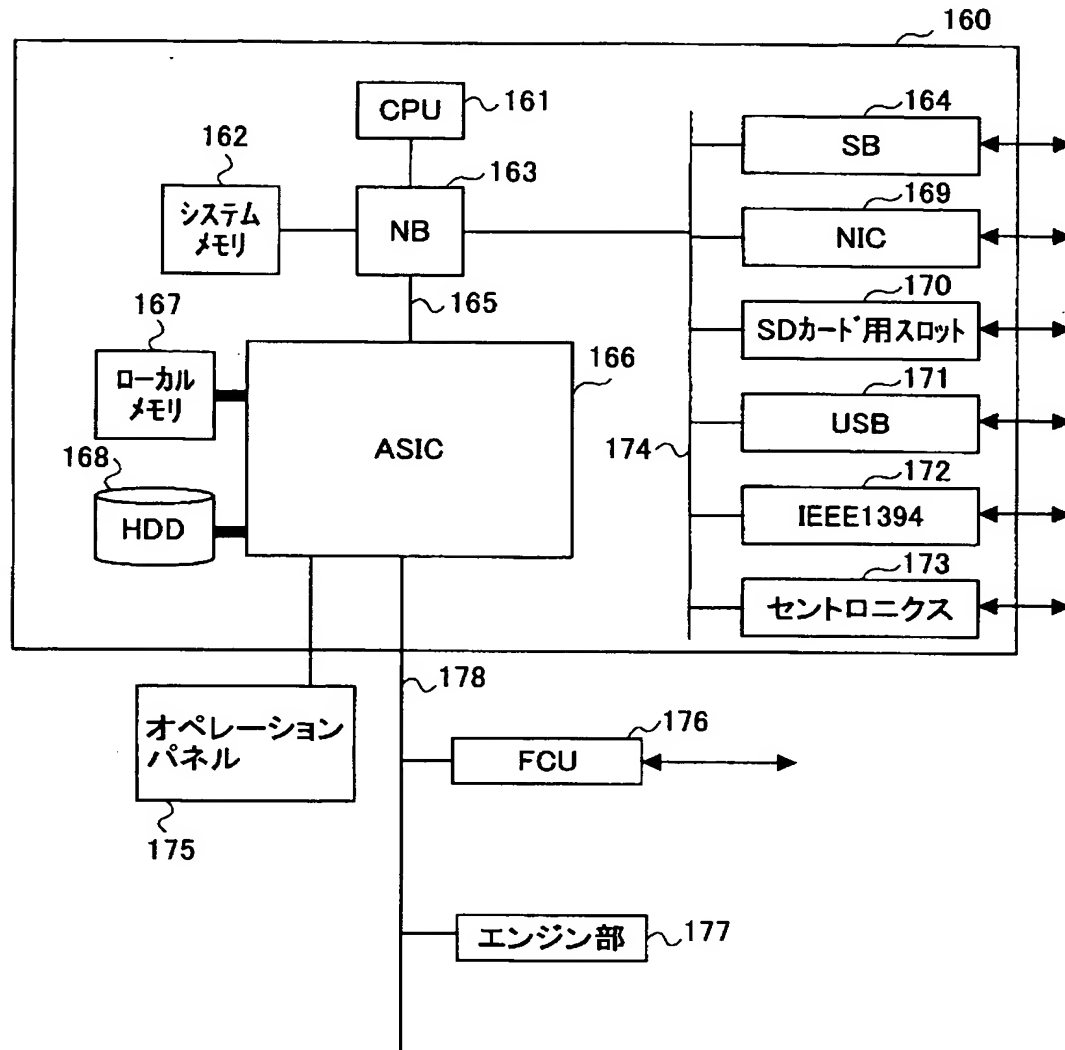
【図 1】

実施の形態 1 にかかる複合機の構成を示すブロック図



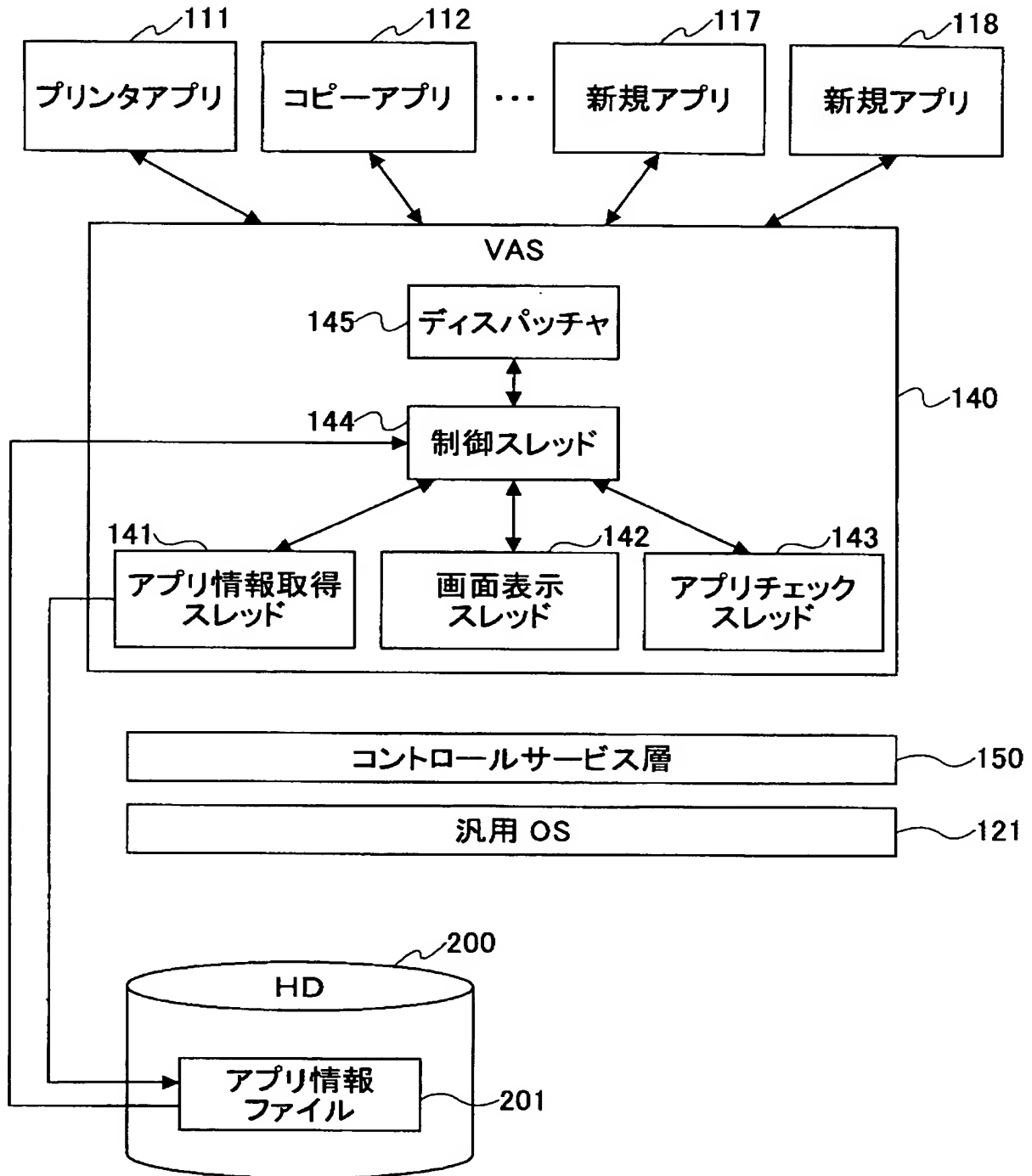
【図2】

実施の形態1にかかる複合機のハードウェア構成を示すブロック図



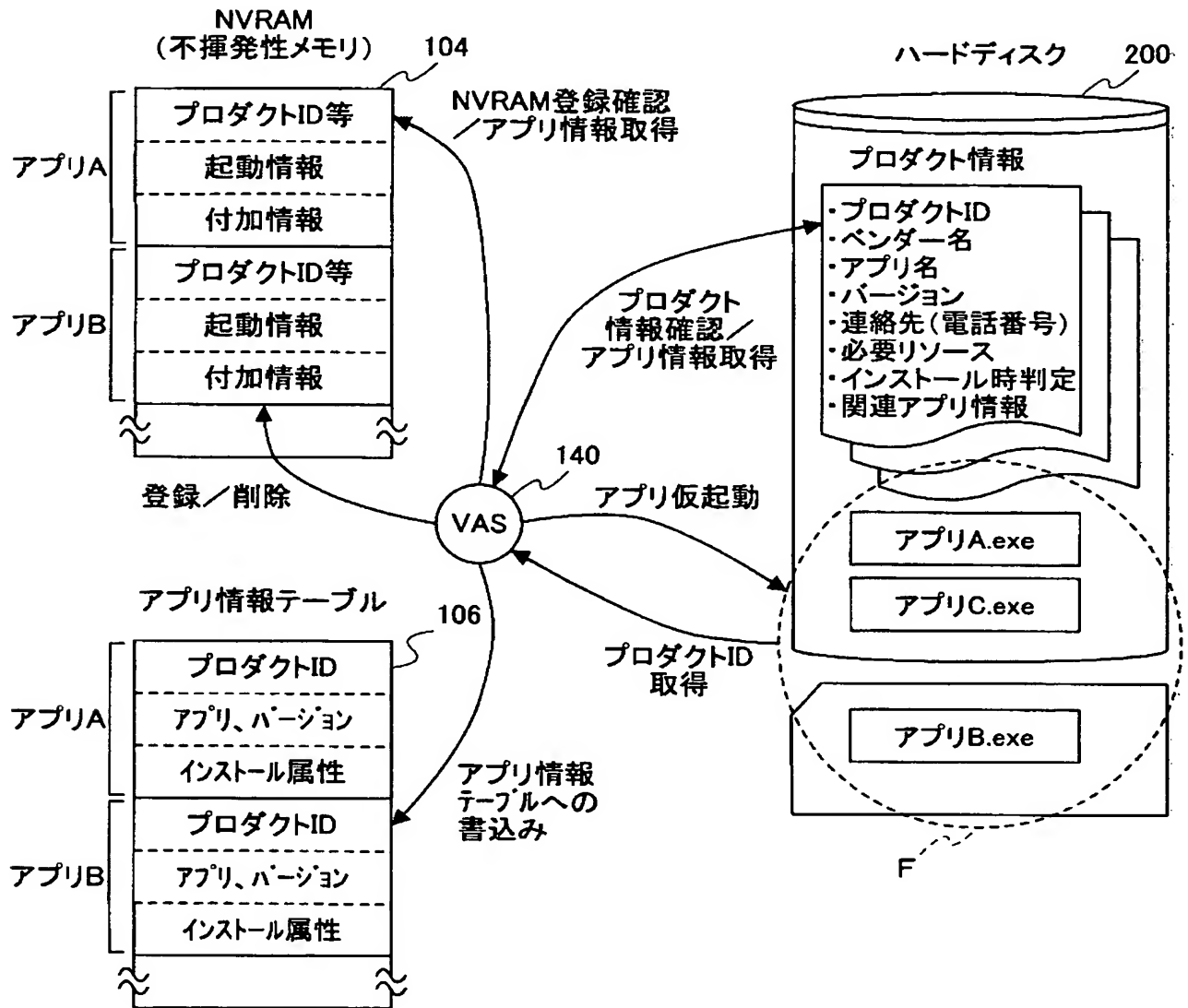
【図 3】

本実施の形態にかかる複合機のVASの構成と、VASと各アプリ、コントロールサービス層および汎用OSとの関係を示すブロック図



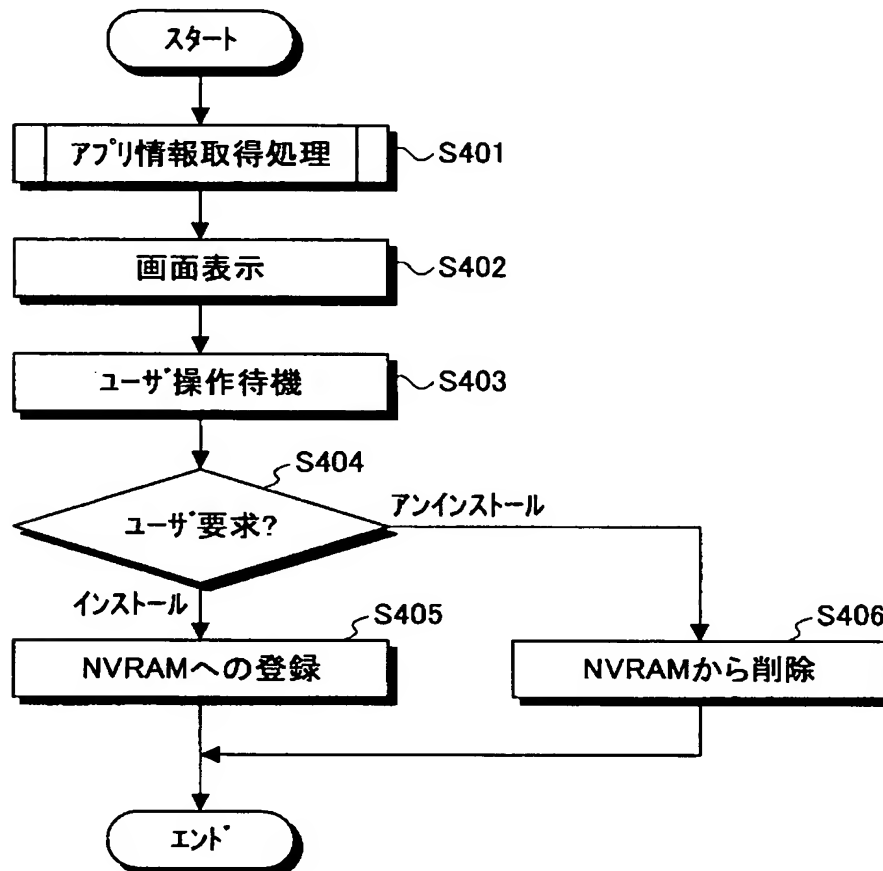
【図4】

VASとアプリ情報を格納するハードディスク、  
アプリ情報テーブル、NVRAMとの関係を示す説明図

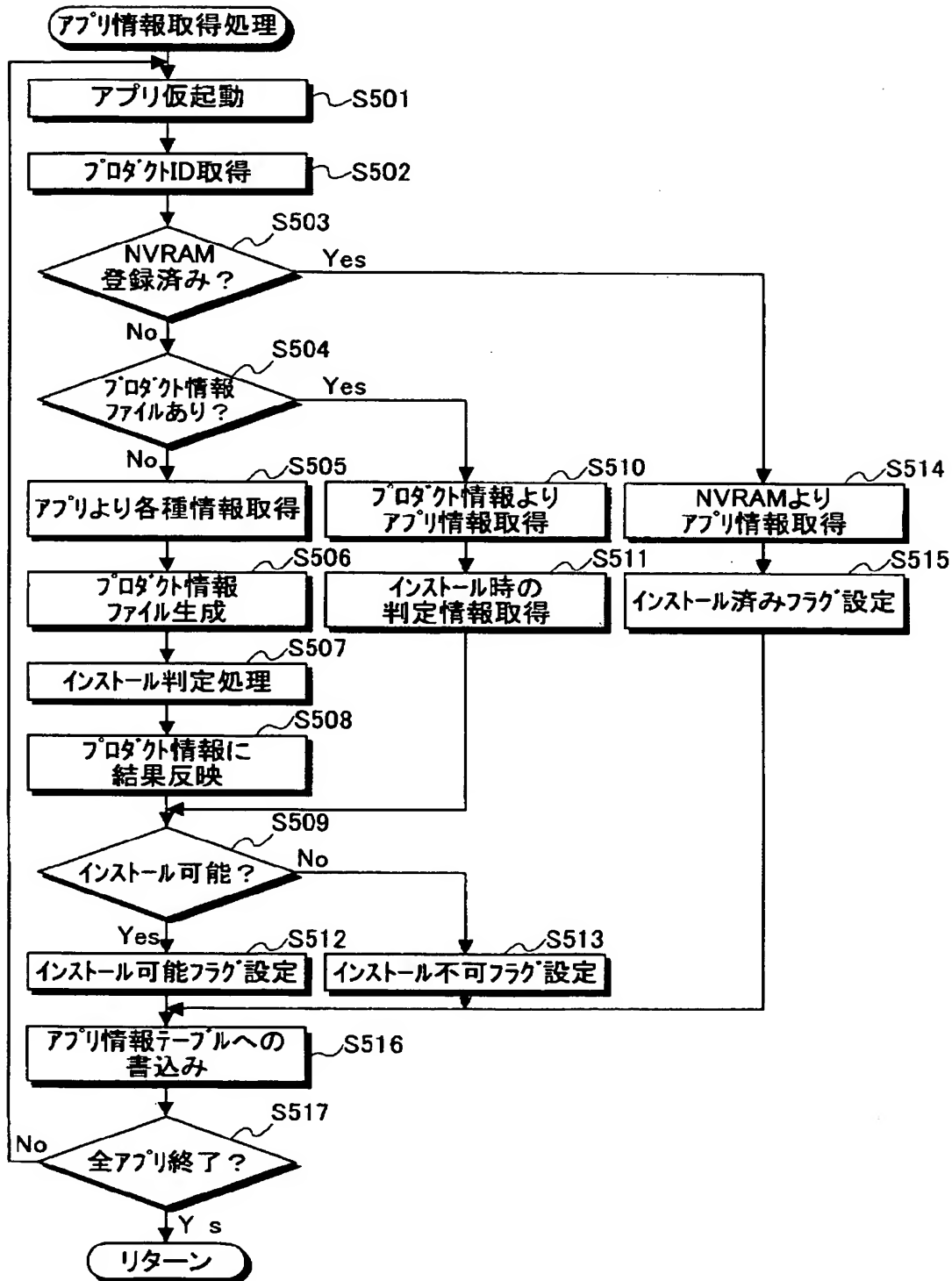


【図 5】

本実施の形態の複合機においてアプリに対してインストール処理  
かアンインストール処理を行う場合の全体フローチャート



【図6】

図5のステップS401におけるアプリ情報取得処理  
のサブルーチンを示すフローチャート

【図 7】

### 仮起動の機能を含むアプリのプログラム記述 (メイン関数)の概要を示す図

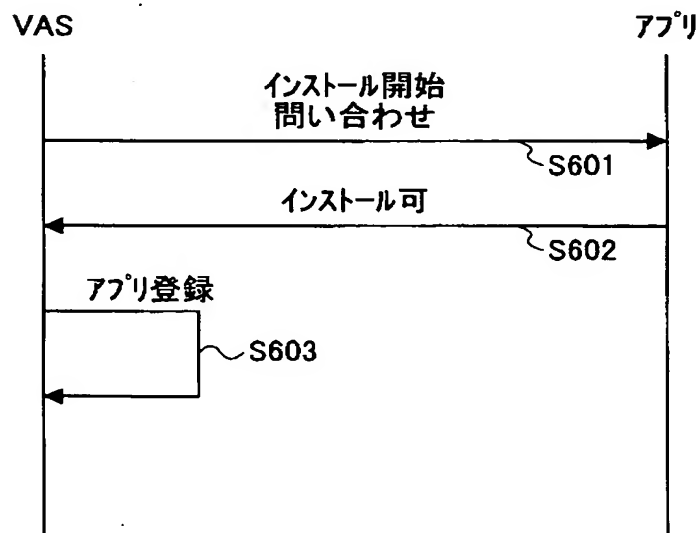
```

int main(int argc, char**argv)
{
    if(argc==2) {
        if(strcmp(argv[1], "-v")==0)
            // 仮起動実行
            if(ConnectVAS()==OK){ // VASとの通信準備
                SetApliInfo(); // アプリ情報提供処理
                Close // 通信終了
            }
        exit(1)
    }
}
// 通常起動処理


```

【図 8】

### VASとアプリのネゴシエーションを示す図







【書類名】 要約書

【要約】

【課題】 インストールが可能なアプリケーションか否かを判断するために必要なアプリケーションに関するアプリ情報を取得する。

【解決手段】 アプリケーションをインストール可能に構成された画像形成装置において、アプリケーションを前記画像形成装置にインストールすることが可能か否かを判断するために使用されるアプリケーション情報の取得を行うアプリケーション情報取得手段を備える。

【選択図】 図 3

特願 2003-317282

出 願 人 履 歴 情 報

識別番号

[000006747]

1. 変更年月日

2002年 5月17日

[変更理由]

住所変更

住 所

東京都大田区中馬込1丁目3番6号

氏 名

株式会社リコー